USDA Service Center Agencies
Geospatial Data Management Team
Data Management Plan For

Climate-Precipitation and Temperature Data

Revised June 2008
Randy Frosh

I. Purpose and Scope (business case)

A. Purpose

The climate information is a product of the Parameter-elevation Regressions on Independent Slopes Model (PRISM). The Spatial Climate Analysis Service of the Oregon Climate Service located at Oregon State University produces the PRISM layers. PRISM incorporates a spatial climate knowledge base that accounts for rain shadows, temperature inversions, coastal effects, and more in the climate mapping process. PRISM data includes:

- Precipitation
- Maximum Temperature
- Minimum Temperature
- Average Temperature
- Dewpoint
- Standardized Precipitation Index
- Percent of Normal Precipitation

Only precipitation and temperature are being used at this time because they are the only theme that is available as a vector layer. All others are raster/grid layers.

The purpose of the precipitation data is for display and/or analyses requiring spatially distributed monthly or annual average precipitation for the climatological period 1971-90. These maps are currently 2.5-minute resolution.

The purpose of the temperature data is for display and/or analyses requiring spatially distributed temperature for the climatological period 1971-2000. These maps are 30 arc-second resolution.

Precipitation will be updated to 1971-2000 and 30 arc-seconds in the near future.

B. Scope

The scope of the dataset is the 48 coterminous states of the United States.

II. Acquisition

A. Data Source

1. Producer Information
   a. Name

      Spatial Climate Analysis Service

b. Location of Headquarters

Strand Agriculture Hall, Room 316
Oregon State University
Corvallis, OR 97331-2209

c. Internet Address

http://www.ocs.orst.edu/prism/

2. Publisher Information

a. Name

Spatial Climate Analysis Service

b. Location of Headquarters

Strand Agriculture Hall, Room 316
Oregon State University
Corvallis, OR 97331-2209

c. Internet Address

http://www.ocs.orst.edu/prism/

3. Acquisition Information

a. Delivery Media

Digital PRISM data sets are available via anonymous File Transfer Protocol (ftp).

b. Download URL

http://www.ocs.orst.edu/prism/state_products/us_maps.html

c. Projected Data Availability Schedule

The data is currently available.

B. Standards Information

1. Geospatial Data Standard

a. Standard Name and Steward Information

United States Department of Agriculture (USDA) Service Center Agencies (SCA) Standard For Geospatial Data

b. Standard Version

SCI Std 003-02
October 15, 2003

c. Standard URL

http://www.itc.nrcs.usda.gov/scdm/docs/SPG-GeospatialDataStandard.pdf

2. Metadata Standard

a. Standard Name and Steward Information

Metadata are compliant with:
Federal Geographic Data Committee (FGDC)
Content Standard for Digital Geographic Metadata FGDC
STD-001-1998 Version 2 revised June 1998

And:
United States Department of Agriculture (USDA) Service Center Agencies (SCA)
Standard For Geospatial Dataset Metadata
SCI Std 003-02 October 15, 2003

http://www.itc.nrcs.usda.gov/scdm/docs/SPG-GeospatialDatasetFileMetadata.pdf

b. Description of Metadata Captured

A metadata text file is distributed with each PRISM data set. The metadata text file provides information on the content, quality, condition, and other characteristics of the data. The sections of metadata include the following:

Identification_Information
Data_Quality_Information
Spatial_Data_Organization_Information
Spatial_Reference_Information
Entity_and_Attribute_Information
Distribution_Information
Metadata_Reference_Information

c. Metadata Accuracy and Completeness Assessment

The metadata is typically complete.

C. Acquired Data Structure

1. Geospatial Data Format

a. Format (raster, vector, etc.)

The format is vector.

b. Format Name

ESRI exchange (.e00)

c. Data Extent

The extent is the 48 contiguous states of the United States.

d. Horizontal and Vertical Resolution

PRISM is being used to model climate elements and derived variables at 30 arc-second horizontal resolution using the latitude-longitude coordinate system

e. Absolute Horizontal and Vertical Accuracy

The absolute horizontal accuracy is 30 arc-seconds.

Accuracy of this data set is based on the original specification of the Defense Mapping Agency (DMA) 1 degree digital elevation models (DEM). The stated accuracy of the original DEMs are 130 m circular error with 90% probability.

Checks were made to ensure that no two adjacent polygons are labeled with the same precipitation value. A manual spot check was made of peaks, depressions, and islands.

f. Nominal Scale

Maps are made using the 1:250,000 scale DEM, and are displayed at scales ranging from 1:250,000 to 1:2 million. Typically scales are 1:1 million.

g. Horizontal and Vertical Datum

Horizontal datum name: North American datum 1983 (NAD83)
Ellipsoid Name: GRS1980
Semi major Axis: 6378137.0

The vertical datum is mean sea level.

h. Projection

Geographic (longitude/latitude)

i. Coordinate Units

Decimal degrees

j. Average Data Set Size

ARC interchanges files average 31 kilobytes to 910 kilobytes.
The metadata averages 21 kilobytes

k. Symbology

None

2. Attribute Data Format

a. Format Name

The attribute data are distributed as part of the .e00 file

b. Database Size

The attribute data are a part of the .e00 file

3. Data Model

a. Geospatial Data Structure

The vector data is incorporated into the .e00 file.

b. Attribute Data Structure

The attribute data are a part of the .e00 file

c. Database Table Definition

None

d. Data Relationship Definition

The means for linking the attributes to the features is maintained using feature items.

e. Data Dictionary

AREA - Area in square degrees
PERIMETER- Distance around polygon in degrees
COV# - Sequential Feature ID
COVID – Sequential coverage feature ID
RANGE - Fixed value of precipitation for the polygons area
LEGEND- Minimum and maximum range of precipitation for the area

D. Policies

1. Restrictions

a. Use Constraints

No restrictions apply

b. Access Constraints

No restrictions apply

c. Certification Issues

For the attribute data derived from PRISM to be used for program activities, the state Climate Data Liaison must certify the data.

Acknowledgement of the following agencies in products derived from these data: Natural Resources Conservation Service (NRCS) Water and Climate Center, NRCS National Cartography and Geospatial Center (NCGC), PRISM Model, and the Oregon Climate Service at Oregon State University.

Point estimates of precipitation originated from the following sources:

- National Weather Service Cooperative (COOP) stations,
- Natural Resources Conservation Service (NRCS) SNOTEL
- Local networks.

The National Climatic Data Center (NCDC) subjected all COOP station data to quality control checks.

2. Maintenance

a. Temporal Information

1971-2000

b. Average -Cycle

Periodic on at least a 10-year update cycle. This data set was updated in 2001 for the new 1971-2000 climatological period.

E. Acquisition Cost

1. Cooperative Agreement

a. Description of Agreement

These data sets have been developed through projects funded partly by the USDA Natural Resources Conservation Service, USDA Forest Service, NOAA Office of Global Programs, and others.

b. Status of Agreement

The agreement is ongoing.

2. Cost to Acquire Data

There is no direct cost.

III. Integration

A. Value Added Process

1. Benefit to the Service Center

A continuous database for the US is created so that any area can be extracted as a shape file. A consistently named FIPS code field is created so that counties and states can easily be extracted.

2. Process Model

a. Flow Diagram
See process description

b. Process Description

East, Central, and West regional .coverage files for created from a series of amls that have been located in \\V480a\gtb1\PRISM\amls_polygons_mrcs, These amls were obtained from http://www.ocs.orst.edu/prism/. The amls and help .html files are also located in PRISM AMLS AND HELP FILES section below.

- Each coverage is then intersected with a TIGER 98 state boundary map.
- The resulting intersect maps are merged into a national shape file using the FIPS_C field from the state map and the RANGE field from the climate map.
- A SDE layer for each month and annual is created for precipitation. A minimum, maximum and average temperature layer is created for temperature.
- The national map is loaded to each SDE layer

For prepping and loading the data onto the Data Gateway:

1. Write or revise the Product Description file (link from "Status Maps" page)
2. Generate shape files for each product with the DataServicesDriver this is located at D:\GatewayWeb\bin\DataservicesDriver.exe using the “All states using current product” option. Check the log file before you go to step 3. Log file location D:Gatewaylogs. Review the log for any errors.
3. Run D:\Gateway\Footprints\CatalogFP_Maker.exe for each product and generate the catalog shape files.

4. Run MakePreviews.exe for each product to generate the preview images for step two of the gateway ordering process. There is no metadata for each map.
5. Create the Status Maps (link from "Status Maps" page) for each product.
6. Move SDE database from staging computer to final SDE database.
7. Edit AXL files for final database location.
8. Notify gateway Fort Collins team to load the catalogs, status maps and news.

3. Technical Issues

a. Tiling

None

b. Compression

None

c. Scale

Same as source

d. Tonal Matching

None

e. Edge-matching

None

4. Quality Control

a. Procedures

Plotting and visual review of SDE database

b. Acceptance Criteria

None

5. Data Steward

a. Name and Organization

National Cartography and Geospatial Center
Natural Resources Conservation Service
US Department of Agriculture
501 Felix Street, Building 23
P. O. Box 6567
Fort Worth, Texas 76115-0567 USA

b. Responsibilities

NCGC is responsible for storage and access of the data.

B. Integrated Data Structure

1. Geospatial Data Format

a. Format (raster, vector, etc.)

Vector

b. Format Name

The format name is ESRI ArcSDE

c. Data Extent

The extent is the 48 coterminous states of the United States

d. Horizontal and Vertical Resolution

The resolution is the same as the source data-2.5 kilometers. The vertical datum is mean sea level.

e. Absolute Horizontal and Vertical Accuracy

The accuracy is the same as the source data- four kilometers.

f. Nominal Scale

The scale is the same as the source data-1:1,000,000

g. Horizontal and Vertical Datum

The datum is the same as the source data-NAD83.

h. Projection

The projection is the same as the source data-geographic.

i. Coordinate Units

The coordinate units are the same as the source data-decimal degrees.

j. Symbology

None

2. Attribute Data Format

a. Format Name

The attributes are part of the SDE layer.

b. Database Size

Part of the SDE layer.

3. Data Model

a. Geospatial Data Structure

ArcSDE DBMS tables

b. Attribute Data Structure

ArcSDE DBMS tables

c. Database Table Definition

ArcSDE DBMS tables

d. Data Relationship Definition

ArcSDE DBMS tables

e. Data Dictionary

The dictionary is the same as the source data and includes Range and Contour

C. Resource Requirements

1. Hardware and Software

This is unknown at this time.

2. Staffing

This is unknown at this time.

D. Integration Cost

1. Hardware and Software

This is unknown at this time.

2. Staffing

This is unknown at this time.

IV. Delivery

A. Specifications

1. Directory Structure

a. Folder Theme Data is Stored In

F:\Geodata
\climate
\precipitation
\temperature

2. File Naming Convention

http://www.itc.nrcs.usda.gov/scdm/docs/SPG-GeospatialDataSetFileNamingStandard.pdf

a. List of Theme Files and The File Naming Convention

Poly Files	
map shp	shp file
map dbf	dbf file
map shx	shx file
map sbn	sbn file
map sbx	sbx file

B. User Information

1. Accuracy Assessment

a. Alignment with Other Theme Geospatial Data

The data is captured at scales varying from 1:250,000 to 1:1 million. There should be some alignment with the ortho-photo layer but this will not be perfect due to the fact that the data is captured at different scales.

b. Content

The mean monthly precipitation estimates were generated by Christopher Daly using the PRISM model (Daly et al. 1994, Daly et al. 1997). Data input to the model consisted of 1971-2000 mean monthly precipitation or temperature from over 8000 NOAA Cooperative sites, SNOTEL sites, and selected state network stations. Data-sparse areas were supplemented by a total of about 500 shorter-term stations. A station was included in this data set if it had at least 20 years of valid data, regardless of its period of record. Patching together individual state maps for the 11 western states, a central US region, an eastern US region, and a New England region made each monthly map. PRISM software was used to minimize "seams" along state and region boundaries.

Summing the 12 monthly maps created the annual maps. The annual maps underwent extensive peer-review by many state climatologists and other experts. This is part of a national effort by the USDA Natural Resources Conservation Service and Oregon State University to develop state-of-the-art precipitation maps for each state in the US, including Alaska and Hawaii.

2. Appropriate Uses of the Geospatial Data

a. Display Scale

The scale is the same as the source. 1:250,000 to 1:2 million. Typically scales are 1:1 million.

b. Plot Scale

Typically scales are 1:1 million.

c. Area Calculations

As accurate as the source data and capture scale and the algorithm used by ArcMap/ArcView

d. Decision Making

As accurate as the source data and capture scale and the algorithm used by ArcMap/ArcView.

C. Maintenance and Updating

1. Recommendations and Guidelines

a. Original data location and structure

The integrated database is at NCGC and the data is delivered to the Service Center.

b. Update Cycle

When the source data is updated.

c. Availability

Make the updates available as soon as the database is updated.

d. Change Control

This is to be determined.

Polygon AML Documentation

1. **Assumptions**
2. **Directory structure explaination**
3. **Step-by-step process for converting a GRID to a POLYGON coverage**
4. **Post-processing GRID to POLYGON transformation**

Assumptions

- You will be logged into account 'jenny' and in directory 'arc'.

Directory structure explaination

- The base directory is /home/typhoon/jenny/arc.
- aml

  This directory contains the aml's you will use. You will eventually have to set the aml path to this directory before running.
- lut

  This directory contains the look-up tables, which are used by our aml's to determine the countour interval and the lable to assign to each polygon. There should be a look-up tables for temperature and precipitation, but in the future you may need to create different ones.
- ppt, tmax, tmin

  These **workspaces** are specific to the type of parameter, like precipitation. There are additional workspaces inside of these that are specific to geographic regions, like 'cent' for central US.

  - <us,west,cent,east>

    us contains the imported GRID for the parent workspace, for example 'ppt/us' would contain the imported precipitation grid for the US.

    west, cent, and east will contain the polygon coverages.

Step-by-step process for converting a GRID to a POLYGON coverage

1. Transfer the ASCII file to the appropriate place.

First change directories to the appropriate location. For example, if you are working on 'tmax', then

cd arc/tmax/us

Now transfer the ASCII file to this location using 'ftp'.

2. Start up ARC/INFO
3. Set the 'amlpath' variable in order for ARC/INFO to find our AML's. Do this by typing

   &amlpath /home/typhoon/jenny/arc/aml

4. Import the ASCII grid with the AML 'ascii2grid_extend.aml'. This will create and ARC/INFO GRID format file that the AML's will operate on.

   &r ascii2grid_extend.aml <ASCII data file name> <mont> <parameter>

   For example: &r ascii2grid_extend.aml us_filt.01.gz 01 tmax

5. Move to the "regional" workspace for the parameter you are processing. For example, to change to the appropriate workspace, type the workspace change command "workspace" ("w" is a shortcut):

   w ../cent

6. Create a subgrid (grid specific to a small region) for the region you are now working in. This will create an ARC/INFO GRID file in the current workspace. Do this by running the AML "subgrid.aml". For example:

   &r subgrid.aml region month parameter

   For example: &r subgrid.aml cent 01 tmax

   where region = west, cent, or east, month = 01-12,14, and parameter = ppt, tmin, or tmax

7. Run the polygon AML "master_extend2.aml". This AML actually runs many other AML's. Simply put, it reads the subgrid created in the previous step, contours, converts the areas between the contour lines into polygons and labels those polygons. Run by typing :

   &r master_extend2.aml <region> <month> <lut>

   where region=west,cent, or east, month = 01-12,14, and lut = lookup table

   For example, &r master_extend2.aml cent 01 us_temp.lut

   This will generate polygon a polygon coverage "cent_01" and sometimes a second *backup* coverage "cent_01_j".

Post-processing GRID to POLYGON transformation

- Check to make sure AML completed

There are 2 ways to determine if the AML ran through completion: a) ther should be no error messages on the screen (trust me ... you will know), and b) inspection of the *log* file , as detailed below.

- General intrepretion of the *log* file

  For each AML run, a log file will be generated with the naming convention of <region>_<month>.log. In the case of problems with the AML, this file will be appended. Here is an example log file that ran through to completion:

  ```
  Opened log file at 98-10-28.14:06:41.Wed
  ----------------------------------------
  Running get_interval.aml
  Getting the cell resolution for setting cutoffs
  Running bound.aml
  Running contour_isl.aml
  Running contour_main.aml
  Intersecting and Joining main_bound and main_poly
  Running polylabel.aml
  Running put_range2.aml on the islands
  Running lookup.aml on the the mainland
  Running lookup.aml on the the islands
  Dropping item edge from the mainland
  Appending the mainland and island coverage together
  Building the entire coverage
  Projecting from WGS72 to NAD83
  Checking for invalid lines
  Invalid lines found 5
  Inspect list in file cent_07.lines
  Closed log file at 98-10-28.14:43:47.Wed
  ```

  The indications that the AML ran correctly are that : a) the coverage was reprojected to NAD83, and b) some "invalid lines" (small line segments) were found. Usually small lines are found and removed but not always. The log file then refers you to the other diagnostics file for inspection.

- General intrepretion of the *lines* file

  More often that not, a *lines* diagnostics file is generated with the naming convention <region>_<month>.lines. This file contains the list of small line segments that were removed. In addition, a second list may be generated that indicates further problem in the polygon labelling process. The specifics of what to look for and how to solve problems wil be discussed next.

- Analysis of the *log/lines* files and modification of the coverage.

  First, if the "_j" version of the coverage doesn't exist, then there were no small line segments to remove. Otherwise, the log file will state that "Invalid lines were found" and the *lines* file will contain a section like the following:

  List of lines being removed from cent_07_j

  Copyright (C) 1982-1997 Environmental Systems Research Institute, Inc.

  All rights reserved.

  ARCEDIT (COGO) Version 7.1.2 (Wed Aug 13 07:45:00 PDT 1997)

  The edit coverage is now /HOME/TYPHOON/JENNY/ARC/TMAX/CENT/CENT_07_J

  Please wait...

  1026 element(s) for edit feature ARC

  Coverage has no COGO attributes

  5 element(s) now selected

  Record $RECNO

  110 110

  113 113

  251 251

  834 834

  915 915

  5 arc(s) deleted

  . .

  . .

  . .

  Checking coverage cent_07.lines

  This output shows that 5 small lines were found and removed. When small lines are found, the coverage is first copied to a "_j" version as a backup and work continues on the non"_j" version. The "_j" version becomes important when larger "dangling arcs" were not removed but are still found in the coverage. At this point, you will have to go back to the "_j" version and manually fix problem arcs by using ARC/INFO. This falls into the "HELP! I'M REALLY LOST" category. However, we rarely need to go back to the "_j" coverage. You can inspect the lines that were removed using ARCVIEW and this is instructive. When you build up confidence in the process, it won't be necessary.

The second set of lines to check are those related to polygon labeling errors. A portion of the log file containing these lines looks like the following

```
Checking coverage cent_07.lines'
Line 21
 lpoly 12
 rpoly 11
 range 82
Line 22
 lpoly 12
 rpoly 13
 range 82
Line 144
 lpoly 78
 rpoly 83
 range 82
Line 474
 lpoly 253
 rpoly 284
 range 94
Line 634
 lpoly 97
 rpoly 392
 range 90
Line 664
 lpoly 411
 rpoly 417
 range 78
```

The next step involves fixing the labels and then running the ARC/INFO command DISSOLVE. The easiest way to check and fix the labels are by using ARCVIEW. ARCVIEW will allow you to change the values of attributes and save them. The list os steps to follow are:

1. Open the *lines* file in a text editor. Notes of your changes to the coverage will be made here and saved.
2. Start ARCVIEW and create a new view.
3. Load the polygon and arcs for the <region>_<month> coverage.
4. Create a "graduated" legend with an appropiate color table.
5. Color the arcs black

6. Turn on both themes and activate the arcs
7. Query the arcs for the 1st line listed in the *lines* file
8. Zoom to the selected arcs and zoom out till you can see some of the adjacent polygons
9. Evaluate whether the adjacent polygons were labelled correctly
10. If the polygon was correctly labelled, then the selected arc will be DISSOLVED in the final processing step
11. If the polygon was labelled incorrectly, determine the polygon id. Then load the polygon attribute table, select the correct polygon, and edit the "range" value replacing it with the correct value.
12. Save the edits and exit the table
13. Repeat steps 7-12 till all suspected arcs have been evaluated.
14. Before proceeding to the DISSOLVE step, you must save and close the coverage in arcview. Do not save the project.

- Final DISSOLVE of the coverage

  If you've made it to this step, then you are just about "home free" Just type the following commands in ARC/INFO

  DISSOLVE <region>_<month> <region>_<month>_dis range net
  KILL <region>_<month> all
  KILL <region>_<month>_j all
  RENAME <region>_<month>_dis <region>_<month>

  Congratulations on a job well done! You are now ready to use this coverage in preparation of hardcopy maps.

- HELP! I'M REALLY LOST (REMEMBER, BE CONSERVATIVE!!)

  See Wayne or Tye. Be aware that for some ASCII grids, the conversion process to polygons can take up to 12 hours of computer time. If you don't understand what you're going to do, ask for help.

Authors: Wayne Gibson and Tye Parzybok

Polygon AML Theory Documentation

General overview

The general steps for converting a grid to a polygon coverage are as follows :

1. Starting with a grid coverage, contour the grid.
2. The contours represent lines where polygons represent areas. We must intersect the contours with some type of boundary coverage to generate polygons.
3. After the intersection has occurred and polygon topology has been built, the polygons must be labelled.
4. After labelling has occurred, small lines must be removed from the data set. Theses are dangling lines, not small polygons.
5. Unfortunately, the process is not foolproof so we run some checks on the labelling process. It is a manual process.
6. If all goes well, an error free topologically correct labelled polygon coverage with labelled contour lines has been generated.

Author: Wayne Gibson

AMLS

1. Master_Extend2.aml

```
/* MASTER_EXTEND2.AML : Creates a polygon coverage from an ASCII grid
for
/* any region.  Calculates interval from user-chosen lut.
/*
/* MUST HAVE IN YOUR WORKSPACE THE GRID: <region>_<month>_g (e.g.
ak_14_g)
/*
/* Choose a lut from /home/array1/lut.
/*
/*  Island_area_size is automatically set to 1
/*
/*  Calls many other amls, the Gaussian filter, reprojects, ...
/*
/* Oregon Climate Service
/* Oregon State University
/* Corvallis, Oregon
/* (c) April 1998
/*
/* Jun 13, 2005 W. Gibson
/* Updating paths of amls, lut, ...
/*
/* Jun 15, 2005 W. Gibson
/* Trying contour_<isl,main>_test.aml versions of AML's
/*
/* Jul 13, 2005 W. Gibson
```

```
/* Created unique ascii file name for the list of contours dumped from
the lut.
/* This should allow for multiple simultaneous runs of this script.
/*
/* Aug 15, 2006 W. Gibson
/* Changed the projection definition to WGS84.
/*****************************************************************
**

/*------------------------------
/* Check command line arguments
/*------------------------------

/*&echo &on

&args region month lut restart

&if [null %region%] &then &do
      &type Usage: MASTER_EXTEND2 <region> <month> <lut> {restart}
      &type Example: MASTER_EXTEND2 ak 14 ak_ann.lut
      &type Note that month must be entered as a 2 character number
(e.g. 01, 02, ... 14)
      &return &warning
&end

&if [null %month%] &then &do
      &type Usage: MASTER_EXTEND2 <region> <month> <lut> {restart}
      &type Example: MASTER_EXTEND2 ak 14 ak_ann.lut
      &type Note that month must be entered as a 2 character number
(e.g. 01, 02, ... 14)
      &return &warning
&end

&if [null %lut%] &then &do
      &type Usage: MASTER_EXTEND2 <region> <month> <lut> {restart}
      &type Example: MASTER_EXTEND2 ak 14 ak_ann.lut
      &type Note that month must be entered as a 2 character number
(e.g. 01, 02, ... 14)
      &return &warning
&end

/*-----------------------------------------------------------------
/* Form file name that fit convention and assign parameters based
/* on cell resolution.
/*-----------------------------------------------------------------

&setv lut_path = /home/array1/lut
&setv prj_dir = /home/array1/prj
&setv work_path = [show workspace]
&setv in_grid = %region%_%month%_g
&setv out_cover = %region%_%month%
&setv isl_bound = isl_bound
&setv isl_grid = isl_grid
&setv main_bound = main_bound
&setv main_grid = main_grid
&setv log_file = %region%_%month%.log
&setv line_file = %region%_%month%.lines
```

```
&setv contour_file = %region%_%month%.contours

&setv isl_area_size = 1

/*-----------------------------------------------------------------
/* Open up a log file to trace where we are in this aml
/* Turn off pagination
/*-----------------------------------------------------------------

&fullscreen &nopaging
&amlpath /home/array1/aml

&setvar fileunit = [open %log_file% openstatus -append]
&setvar start = [date -full]
&setvar success = [write %fileunit% 'Opened log file at '%start%]
&setvar success = [WRITE %fileunit% '---------------------------------
----------']

/*---------------------------------
/* Transfer if rechecking the lines
/*---------------------------------

&if ^ [null %restart%] &then
 &do

   &if ^ [exists %out_cover%_j -network] &then &do
      &type ERROR %out_cover%_j does not exist and it must.
      &type Fix the problem and restart the AML.
      &setvar success = [close %fileunit%]
      &return &warning
   &end

  &severity &error &fail
  &type Rechecking the lines.
  &setvar success = [write %fileunit% 'Rechecking the lines.']
  &goto recheck
 &end

/*-----------------------------------------------------------------
/* Get the interval to use for LatticeContour
/*-----------------------------------------------------------------

&if ^ [exists %lut_path%/%lut% -info] &then &do
  &type Lookup Table does not exist.
  &type Create one for your coverage in /home/array1/lut
  &return &warning
&end

&setvar success = [write %fileunit% '---- Running get_interval.aml ----
']
&type '---- Running get_interval.aml ----'

&r get_interval.aml %lut% %work_path% %contour_file%

/*-----------------------------------------------------------------
/* Get cell resolution from which most of the tolerances are calculated
/*-----------------------------------------------------------------
```

```
&setvar success = [write %fileunit% '---- Getting the cell resolution
for setting cutoffs ----']
&type '---- Getting the cell resolution for setting cutoffs ----'

GRID
setcell %in_grid%
&set min_length = [show setcell]
QUIT

&setv cellres = %min_length%
&setv extend_length = 2 * %min_length%

/*-----------------------------------------------------------------
/* Process raw grid file into mainland and island area grids, create
boundaries
/* for islands and mainland.  ITEM "range" is defined and set to 0 for
water
/* and 1 for land.  This is later changed to ITEM "mask".  Then "mask"
is
/* used to determine whether a polygon should be assigned a missing
value for
/* over water, or a range value calculated for over land.
/*
/* bound_extend.aml <in_grid> <isl_area_size>
/*    <in_grid> is input raster grid
/*    <isl_area_size> maximum area which defines what an island is
/*-----------------------------------------------------------------

&setvar success = [write %fileunit% '---- Running bound.aml ----']
&type '---- Running bound.aml ----'

&r bound.aml %in_grid% %isl_area_size%

/*-----------------------------------------------------------------
/* Contour the island and mainland separately and generate polygon
topology
/*
/* contour.aml <in_grid> <in_bound> <out_coverage> <.contour_interval>
<min_length> <extend_length> <ascii contour file>
/*    <in_grid> = input grid to contour
/*    <in_bound> = boundary for intersection
/*    <out_coverage> = output polygons
/*    <.contour_interval> = contour interval
/*    <min_length> = minimum arc length allowable
/*    <extend_length> = minimum arc length allowable
/*    <ascii contour file> = ascii file that contains the list of
contours
/*-----------------------------------------------------------------

&if [exists isl_bound -network] &then &do
   &setvar success = [write %fileunit% '---- Running contour_isl.aml --
--']
   &type '---- Running contour_isl.aml ----'
   &type The contour interval is %.contour_interval%
   &r contour_isl_test.aml isl_grid isl_bound isl_poly
%.contour_interval% %min_length% %extend_length% %contour_file%
```

```
/*   &r contour_isl.aml isl_grid isl_bound isl_poly %.contour_interval%
%min_length% %extend_length%
&end

&setvar success = [write %fileunit% '---- Running contour_main.aml ----
']
&type '---- Running contour_main.aml ----'

&type The contour interval is %.contour_interval%
/*&r contour_main.aml main_grid main_bound main_poly
%.contour_interval% %min_length%
&r contour_main_test.aml main_grid main_bound main_poly
%.contour_interval% %min_length% %contour_file%

/*-----------------------------------------------------------------
/* Intersect main_bound with main_poly
/* This is where the ITEM "range" is originally set for the polygons
/* and then renamed to "mask".  "polylabel.aml" takes advantage of this
/* in determining whether to calculate a value "range" or set it to
/* missing.
/*-----------------------------------------------------------------

&setvar success = [write %fileunit% '---- Intersecting and Joining
main_bound and main_poly ----']
&type '---- Intersecting and Joining main_bound and main_poly ----'

INTERSECT main_bound main_poly junk1 poly .00001 join
JOINITEM main_poly.pat junk1.pat main_poly.pat main_poly-id main_poly-
id
kill junk1 all
DROPITEM main_poly.pat main_poly.pat junk1#
DROPITEM main_poly.pat main_poly.pat junk1-id
DROPITEM main_poly.pat main_poly.pat main_bound#
DROPITEM main_poly.pat main_poly.pat main_bound-id

tables
sel main_poly.pat
alter range,mask,,,,,
quit

/*-----------------------------------------------------------------
/* Create the labels for each of the polygons.  Mainland and island
each
/*  use different amls
/*
/* polylabel.aml <coverage>
/*-----------------------------------------------------------------

&setvar success = [write %fileunit% '---- Running polylabel.aml ----']
&type '---- Running polylabel.aml ----'

&r polylabel.aml main_poly

/*** Temporary stop ***
/*&setvar success = [close %fileunit%]
/*&return
/****
```

```
&if [exists isl_poly -network] &then &do
   &setvar success = [write %fileunit% '---- Running put_range2.aml on
the islands ----']
   &type '---- Running put_range2.aml on the islands ----'
   &r put_range2.aml isl_poly isl_grid %cellres%
&end

/*-----------------------------------------------------------------
/* Run range values through lookup table
/* drop the item "edge" as it is no longer needed.
/*-----------------------------------------------------------------

&setvar success = [write %fileunit% '---- Running lookup.aml on the the
mainland ----']
&type '---- Running lookup.aml on the the mainland ----'

&r lookup.aml main_poly %lut_path%/%lut%

&if [exists isl_poly -network] &then &do
   &setvar success = [write %fileunit% '---- Running lookup.aml on the
the islands ----']
   type '---- Running lookup.aml on the the islands ----'
   &r lookup.aml isl_poly %lut_path%/%lut%
&end

&setvar success = [write %fileunit% '---- Dropping item edge from the
mainland ----']
&type '---- Dropping item edge from the mainland ----'

dropitem main_poly.pat main_poly.pat edge

&setvar success = [write %fileunit% '---- Appending the mainland and
island coverage together ----']
&type '---- Appending the mainland and island coverage together ----'

/*-----------------------------------------------------------------
/* Append the islands and the mainland coverage.  Then build.
/*-----------------------------------------------------------------

DROPITEM main_poly.pat main_poly.pat mask

&if [exists isl_poly -network] &then &do
   APPEND junk1 net
   main_poly
   isl_poly
   END
   &setvar success = [write %fileunit% '---- Building the entire
coverage ----']
   &type '---- Building the entire coverage ----'
   build junk1
&end

&else
  rename main_poly junk1

/*/*-----------------------------------------------------------------
```

```
/*/* Reproject to NAD83
/*/*-----------------------------------------------------------------
/*
/*&setvar success = [write %fileunit% '---- Projecting from WGS72 to
NAD83 ----']
/*&type '---- Projecting from WGS72 to NAD83 ----'
/*
/*project cover junk1 junk2 %prj_dir%/wgs72to84.prj
/*project cover junk2 %out_cover% %prj_dir%/wgs84tonad83.prj
/*build %out_cover%
/*
/* Change the datum name from nar_c to nad83.
/*
/*projectdefine cover %out_cover%
/*datum nad83
/*parameters

/* No need to reproject

rename junk1 %out_cover%
build %out_cover%

projectdefine cover %out_cover%
datum WGS84
parameters


/*-----------------------------------------------------------------
/* Cleanup intermediate files
/* Rename cover to "coverage"_j just in case the deletion of arcs was
/* a mistake.  This allows one to recover from the error.  if the "_j"
/* cover doesn't exist, then no lines were deleted.
/*-----------------------------------------------------------------
&severity &error &ignore

kill junk1 all
kill junk2 all
kill main_poly all
kill isl_poly all
kill main_bound all
kill isl_bound all
kill isl_grid all
kill main_grid all

&severity &error &fail

rename %out_cover% %out_cover%_j

&label recheck

/*-----------------------------------------------------------------
/* Get rid of arcs that don't form polygon topology
/*-----------------------------------------------------------------

&setvar success = [write %fileunit% '---- Checking for invalid lines --
--']
&type '---- Checking for invalid lines ----'
```

```
ARCPLOT
aselect %out_cover%_j line
reselect %out_cover%_j line lpoly# = rpoly#
&sv nrec := [extract 1 [show select %out_cover%_j line]]
QUIT

&if %nrec% >= 1 &then &do
&setvar success = [write %fileunit% 'Invalid lines found '%nrec%]
&setvar success = [write %fileunit% 'Inspect list in file '%line_file%]
&type Found %nrec% invalid lines

&watch %line_file% &append
&type List of lines being removed from %out_cover%_j

ARCEDIT
edit %out_cover%_j
ef arc
select rpoly# = lpoly#
list $recno
delete
build
save %out_cover%
QUIT
&watch &off
&end

&else
rename %out_cover%_j %out_cover%

/*-----------------------------------------------------------------
/* Run the final check that for polygons adjacent to each line, the
range
/* value is different.  If not, write out the line and polygons to a
log file.
/*-----------------------------------------------------------------

&r check_line.aml %out_cover%

&setvar ending = [date -full]
&setvar success = [write %fileunit% 'Opened log file at '%start%]
&setvar success = [write %fileunit% 'Closed log file at '%ending%]
&setvar success = [close %fileunit%]

/*&echo &off

&return
```

2. Check_line.aml

```
/*****************************************************************
*
/* CHECK_LINE - tests that no 2 adjacent polygons have the same RANGE
/* value. Lists the line id and adjacent polygon id and ranges.
/*
/* Oregon Climate Service
/* Oregon State University
/* Authors: W. Gibson, J. Weisberg
```

```
/* (c) March 1998
/*
/* 10/28/98 W. Gibson, T. Parzybok
/* Changed missing value from -999 to -9999
/*****************************************************************
*

&args .cp$cover

ARCPLOT
/*****************************************************************
/* check for required arguments and valid 'NET' (line and poly) cover
/*****************************************************************

&if [null %.cp$cover%] &then &do
  &type Usage: CHECK_LINE <in_cover>
  &return &warning
&end

&if ^ [exists %.cp$cover% -network] &then &do
  &type Input cover not found or doesn't contain both line and poly
attributes
  &return &warning
&end

/*********************************
/* find number of lines in cover
/*********************************

&setvar missing = -9999
ASELECT %.cp$cover% line
RESELECT  %.cp$cover% line contour > %missing%

/*&echo &on

&sv nrec := [extract 1 [show select %.cp$cover% line]]

&type Found a possible %nrec% lines to check

cursor fixcur declare %.cp$cover% line rw
cursor fixcur open

/* --------- MAIN LOOP -------------

&sv filename = %.cp$cover%.lines
&sv fileunit [open %filename% openstatus -append]
&if %openstatus% <> 0 &then &do
      &type Unsuccessful opening of %filename%
      &type Script exiting
      &return &warning
&end

&setvar success = [WRITE %fileunit% 'Checking coverage '%filename%']

&do &while %:fixcur.aml$next%
        &setv i = %:fixcur.$recno%
```

```
&type Working on record number %i%

/* ------ Get the current polygon range value

ASELECT %.cp$cover% line
RESELECT %.cp$cover% line $recno = %i%

  STATISTICS %.cp$cover% line
    max lpoly#
    max rpoly#
  end

  &setvar lpol = [show statistic 1 1]
  &setvar rpol = [show statistic 2 1]

&if %rpol% <> 1 and %lpol% <> 1 &then &do

ASELECT %.cp$cover% poly
RESELECT %.cp$cover% poly $recno = %lpol%

  STATISTICS %.cp$cover% poly
    max range
  end
  &setvar lpol_range = [show statistic 1 1]

ASELECT %.cp$cover% poly
RESELECT %.cp$cover% poly $recno = %rpol%

  STATISTICS %.cp$cover% poly
    max range
  end
  &setvar rpol_range = [show statistic 1 1]

  &if %rpol_range% = %lpol_range% &then &do
    &setvar success = [WRITE %fileunit% 'Line '%i%]
    &setvar success = [WRITE %fileunit% ' lpoly '%lpol%]
    &setvar success = [WRITE %fileunit% ' rpoly '%rpol%]
    &setvar success = [WRITE %fileunit% ' range '%rpol_range%]
  &end

&end
cursor fixcur next

/* end of main loop for PASS 2

&end /*

cursor fixcur close
cursor fixcur remove
&setvar success = [close %fileunit%]

QUIT
&return
```

3. Bound.aml

```
/*************************************************************************
*
/* BOUND - Generates a new grid coverage which has NODATA set for
values
/* less than -9999, creates the land sea mask (polygons).
/*
/*
/* Oregon Climate Service
/* Oregon State University
/* Authors: W. Gibson, J. Weisberg
/* (c) March 1998
/*
/* May 6, 1998
/* Changed the way we get the island vs mainland borders to a more
/* robust algorithm.
/*
/* 10/28/98 W. Gibson, T. Parzybok
/* Changed missing value from -999 to -9999
/*
/* 6/27/05 W. Gibson
/* Rewrote method of creating boundaries.  Boundaries should now be
/* 1/2 grid cell from extremities.  Contours should stop at or beyond
/* the boundary.
/*
/* 7/15/05 W. Gibson
/* Changed BUILD to CLEAN after LATTICECONTOUR of main_01 and isl_01

&args .cp$cover .cp$area

/*&if [show program] ne GRID &then &do
/*  &type This program must be run from GRID
/*  &return &warning
/*&end

&setv out_grid = temp_grid
&setv gtemp = temp_grid2
&setv mask = temp_mask
&setv mask_bound = mask_bound
&setv main_bound = main_bound
&setv isl_bound = isl_bound

/*************************************************************************
/* check for required arguments and valid 'NET' (line and poly) cover
/*************************************************************************

&if [null %.cp$cover%] &then &do
  &type Usage: BOUND <in_grid> <area>
  &return &warning
&end

&if [null %.cp$area%] &then &do
  &type Usage: BOUND <in_grid> <area>
  &return &warning
&end
```

```
&if ^ [exists %.cp$cover% -grid] &then &do
  &type Input cover not found or doesn't contain both line and poly
attributes
  &return &warning
&end

&if [exists %out_grid% -grid] &then &do
  &type Output cover found
  &return &warning
&end

&if [exists %main_bound% -network] &then &do
  &type Output cover found
  &return &warning
&end

&if [exists %mask% -network] &then &do
  &type MASK cover found
  &return &warning
&end


/*************************************************************
/* set window, cellsize, expand window, and snap expanded window
/* to original grid.
/*************************************************************

GRID
setwindow %.cp$cover%
setcell %.cp$cover%

&set cell = [show setcell]
&set swind = [show setwindow]

&set w = [extract 1 %swind%]
&set s = [extract 2 %swind%]
&set e = [extract 3 %swind%]
&set n = [extract 4 %swind%]

&set wc = %w% - %cell%
&set sc = %s% - %cell%
&set ec =  %e% + %cell%
&set nc = %n% + %cell%

&type Expanding the grid on all sides by %cell%
setwindow %wc% %sc% %ec% %nc% %.cp$cover%

/*************************************************************
/* Create output grid with NODATA values and then create polygon
/* coverage of land/water mask.
/*************************************************************


&type Setting NODATA value and creating land/water grid

%out_grid% = setnull (%.cp$cover% <= -9999, %.cp$cover%)
if (%out_grid%) then
```

```
   junk = 1
endif

%mask% = int(junk)
kill junk all
QUIT

gridpoly %mask% %mask_bound%

&severity &error &ignore
ARCEDIT
edit %mask_bound%
editfeature arc
drawenvironment arc
select lpoly# = rpoly#
delete
build
save
QUIT
&severity &error &fail

/* Create masks for main and isl that encompass the same land/sea
extent

ARCEDIT
edit %mask_bound%
ef poly
sel area < %.cp$area%
put isl_mask
delete
save main_mask
QUIT

build isl_mask poly
build main_mask poly
build isl_mask arc
build main_mask arc

/* Convert polygons to grids
GRID
setwindow %wc% %sc% %ec% %nc% %.cp$cover%
setcell %.cp$cover%
isl_mgrid = polygrid(isl_mask,grid-code,#,#,#)
main_mgrid = polygrid(main_mask,grid-code,#,#,#)
QUIT

/* Extract the grids for the mainland and island

GRID
SETWINDOW isl_mgrid
SETCELL isl_mgrid

isl_grid = con(isl_mgrid >= 1,%.cp$cover%)
isl_01 = con(isnull(isl_mgrid),0,isl_mgrid)

SETWINDOW main_mgrid
SETCELL main_mgrid
```

```
main_grid = con(main_mgrid >= 1,%.cp$cover%)
main_01 = con(isnull(main_mgrid),0,main_mgrid)

QUIT

/* Create island and main boundary

LATTICECONTOUR isl_01 %isl_bound% 10 1 # 0.000001
LATTICECONTOUR main_01 %main_bound% 10 1 # 0.000001
/*BUILD %isl_bound%
/*BUILD %main_bound%
CLEAN %isl_bound% %isl_bound% # 0.000001
CLEAN %main_bound% %main_bound% # 0.000001

&severity &error &ignore
ARCEDIT
edit %main_bound%
editfeature arc
drawenvironment arc
select all
clean .00001
select lpoly# = rpoly#
delete
build
save
select all
calc contour = -9999
save

/* Set contour to -9999
edit %isl_bound%
editfeature arc
drawenvironment arc
select all
clean .00001
select lpoly# = rpoly#
delete
build
save
select all
calc contour = -9999
save
QUIT
&severity &error &fail


/*************************************************************
/* Assign Polygon label for land/water
/*************************************************************

&r put_range2.aml %main_bound% main_01 %cell%
&r put_range2.aml %isl_bound% isl_01 %cell%

/*************************************************************
/* Cleanup
/*************************************************************
```

```
&severity &error &ignore

/* TEST

/*&severity &error &fail
/*&return

/* END TEST

kill %out_grid% all
kill %mask% all
kill %mask_bound% all
kill isl_01 all
kill main_01 all
kill main_mgrid all
kill isl_mgrid all
kill isl_mask all
kill main_mask all

&severity &error &fail

&return
```

4. Contour_isl.aml

```
/*****************************************************************
*
/* CONTOUR - Contours the raster grids at specified interval
/*
/* Oregon Climate Service
/* Oregon State University
/* Authors: W. Gibson, J. Weisberg
/* (c) March 1998
/*
/* 10/28/98 W. Gibson, T. Parzybok
/* Changed missing from -999 to -9999
/* 6/21/05 W. Gibson
/* Added code to ensure that item CONTOUR is defined as floating point.
/* Changed fuzzy tolerance to 1/20 .cp$min_length
/*
/* Jul 13, 2005 W. Gibson
/* Passing file name that contains the list of contour intervals
/*****************************************************************
/* Arguments :
/* .cp$cover = input grid to contour
/* .cp$bound = input boundaries for intersecting
/* .cp$out = output polygon coverage
/* .cp$interval = contour interval
/* .cp$min_length = minimum arc length allowable
/* .cp$extend = length to extend
/*  confile = ascii file that contains the contour intervals

&args .cp$cover .cp$bound .cp$out .cp$interval .cp$min_length
.cp$extend confile

/*&setv fuzzy = .00001
```

```
&setv fuzzy = %.cp$min_length% / 20.0
&type Fuzzy tolerance set to %fuzzy%

/*****************************************************************
/* check for required arguments and valid 'NET' (line and poly) cover
/*****************************************************************

&if [null %.cp$cover%] &then &do
  &type Usage: CONTOUR <in_grid> <in_bound> <out_coverage>
<contour_interval> <min_length> <extend_length> <ascii contour file>
  &return &warning
&end

&if [null %.cp$bound%] &then &do
  &type Usage: CONTOUR <in_grid> <in_bound> <out_coverage>
<contour_interval> <min_length> <extend_length> <ascii contour file>
  &return &warning
&end

&if [null %.cp$out%] &then &do
  &type Usage: CONTOUR <in_grid> <in_bound> <out_coverage>
<contour_interval> <min_length> <extend_length> <ascii contour file>
  &return &warning
&end

&if [null %.cp$interval%] &then &do
  &type Usage: CONTOUR <in_grid> <in_bound> <out_coverage>
<contour_interval> <min_length> <extend_length> <ascii contour file>
  &return &warning
&end

&if [null %.cp$min_length%] &then &do
  &type Usage: CONTOUR <in_grid> <in_bound> <out_coverage>
<contour_interval> <min_length> <extend_length> <ascii contour file>
  &return &warning
&end

&if [null %.cp$extend%] &then &do
  &type Usage: CONTOUR <in_grid> <in_bound> <out_coverage>
<contour_interval> <min_length> <extend_length> <ascii contour file>
  &return &warning
&end

&if [exists %.cp$out% -network] &then &do
  &type Output cover found
  &return &warning
&end

&if ^ [exists %.cp$cover% -grid] &then &do
  &type Input cover missing
  &return &warning
&end

&if ^ [exists %.cp$bound% -network] &then &do
  &type Input boundary cover missing
  &return &warning
&end
```

```
&if ^ [exists %confile%] &then &do
  &type ASCII file name that contains the list of contours is missing
  &return &warning
&end

/* Contour the grid, go into arcedit and clean/build/ ...
&severity &error &ignore

latticecontour %.cp$cover% %.cp$out% %.cp$interval% 0 # %fuzzy%

/***if isl_poly exists, make sure CONTOUR is defined as float

&if [exists %.cp$out% -arc] &then &do
  TABLES
  ADDITEM %.cp$out%.aat contour2 4 12 F 3
  SELECT %.cp$out%.aat
  CALC contour2 = contour
  DROPITEM %.cp$out%.aat contour
  alter contour2,contour,,,,,
  COMMIT
  QUIT
&end

/***if isl_poly doesn't exist, copy isl_bound cover to isl_poly

&if ^ [exists %.cp$out% -arc] &then &do
  copy %.cp$bound% %.cp$out%
  tables
  sel %.cp$out%.aat
  alter contour,contour,12,f,3,,
  calc contour = -9999
  quit
&end


/***isl_poly exists
&else &do
&r select_contour.aml %.cp$out% %confile%

ARCEDIT
edit %.cp$out%
ef arc
de arc
sel all
/*unsplit none
/*select length <  %.cp$min_length%
/*delete
nodesnap off
get %.cp$bound%
unsplit none
clean %fuzzy% %.cp$min_length%
select dangle
extend %.cp$extend%
clean %fuzzy% %.cp$min_length%
build
save
```

```
/* Code added to eliminate the interior lines of -9999 and relabel all
/* lines adjacent to the universal polygon with a -9999.  Since lines
/* are deleted, labels also need to be recreated.

select all
select rpoly# <> 1 and lpoly# <> 1
reselect contour = -9999
delete
build
save

select all
reselect rpoly# = 1 || lpoly# = 1
calc contour = -9999
build
save

ef label
sel all
delete
build
save

QUIT
&severity &error &fail

&end
```

5. Contour_main.aml

```
/*****************************************************************
*
/* CONTOUR - Contours the raster grids at specified interval
/*
/* Oregon Climate Service
/* Oregon State University
/* Authors: W. Gibson, J. Weisberg
/* (c) March 1998
/*
/* May 7, 2002 W. Gibson
/* Set fuzzy tolerace to be the same as .cp$min_length.  This helps
/* in extending dangles and small arc removal
/*
/* Small arcs are being removed that should not be.  In the future
/* try and eliminate small arcs by eliminating small areas.
/*
/* Jun 20, 2005 W. Gibson
/* ALTER the definition of CONTOUR to a floating point to be compatable
/* with island polygons
/*
/* Jul 13, 2005 W. Gibson
/* Passing file name that contains the list of contour intervals
/*****************************************************************
/* Arguments :
/* .cp$cover = input grid to contour
```

```
/* .cp$bound = input boundaries for intersecting
/* .cp$out = output polygon coverage
/* .cp$interval = contour interval
/* .cp$min_length = minimum arc length allowable
/*  confile = ascii file that contains the contour intervals

&args .cp$cover .cp$bound .cp$out .cp$interval .cp$min_length confile

&severity &error &ignore

/*&setv fuzzy = .00001
&setv fuzzy = %.cp$min_length% / 200.0
&type Fuzzy tolerance set to %fuzzy%

/*****************************************************************
/* check for required arguments and valid 'NET' (line and poly) cover
/*****************************************************************

&if [null %.cp$cover%] &then &do
  &type Usage: CONTOUR <in_grid> <in_bound> <out_coverage>
<contour_interval> <min_length> <ascii contour file>
  &return &warning
&end

&if [null %.cp$bound%] &then &do
  &type Usage: CONTOUR <in_grid> <in_bound> <out_coverage>
<contour_interval> <min_length> <ascii contour file>
  &return &warning
&end

&if [null %.cp$out%] &then &do
  &type Usage: CONTOUR <in_grid> <in_bound> <out_coverage>
<contour_interval> <min_length> <ascii contour file>
  &return &warning
&end

&if [null %.cp$interval%] &then &do
  &type Usage: CONTOUR <in_grid> <in_bound> <out_coverage>
<contour_interval> <min_length> <ascii contour file>
  &return &warning
&end

&if [null %.cp$min_length%] &then &do
  &type Usage: CONTOUR <in_grid> <in_bound> <out_coverage>
<contour_interval> <min_length> <ascii contour file>
  &return &warning
&end

&if [exists %.cp$out% -network] &then &do
  &type Output cover found
  &return &warning
&end

&if ^ [exists %.cp$cover% -grid] &then &do
  &type Input cover missing
  &return &warning
&end
```

```
&if ^ [exists %.cp$bound% -network] &then &do
  &type Input boundary cover missing
  &return &warning
&end

&if ^ [exists %confile%] &then &do
  &type ASCII file name that contains the list of contours is missing
  &return &warning
&end

/* Contour the grid, go into arcedit and clean/build/ ...

latticecontour %.cp$cover% %.cp$out% %.cp$interval% # # %fuzzy%

&r select_contour.aml %.cp$out% %confile%

ARCEDIT
edit %.cp$out%
ef arc
de arc
sel all
/*unsplit none

/* Jun 23, 2005 W. Gibson.
/* temporarily comment out small arc removals
select length <  %.cp$min_length%
reselect tnode# <> fnode#
delete

nodesnap off
get %.cp$bound%
unsplit none
clean %fuzzy% %.cp$min_length%
&severity &error &ignore
sel dangle
extend %.cp$min_length%
select dangle
delete
&severity &error &fail
build
save

/* Code added to eliminate the interior lines of -9999 and relabel all
/* lines adjacent to the universal polygon with a -9999.  Since lines
/* are deleted, labels also need to be recreated.
/* remove temporarily for debugging. 2/1/2001 W. Gibson

/*select all
/*select rpoly# <> 1 and lpoly# <> 1
/*reselect contour = -9999
/*delete
/*build
/*save
/*
/*select all
/*reselect rpoly# = 1 || lpoly# = 1
```

```
/*calc contour = -9999
/*build
/*save

ef label
sel all
delete
build
save

QUIT

/* ALTER definition of item CONTOUR.
/* Cannot change from a B to a F definition by ALTER alone.

TABLES
ADDITEM %.cp$out%.aat contour2 4 12 F 3
SELECT %.cp$out%.aat
CALC contour2 = contour
DROPITEM %.cp$out%.aat contour
alter contour2,contour,,,,,
COMMIT
QUIT
```

6. lookup.aml

```
/*
/* LOOKUP : Converts the Range item to RANGE-CODE in the lookup table.
/*
/* Oregon Climate Service
/* Oregon State University
/* Authors: W. Gibson, J. Weisberg
/* (c) March 1998
/*

&args poly_cov lut

&if [null %poly_cov%] &then &do
  &type Usage: LOOKUP <poly_coverage> <lookup_table>
  &return &warning
&end

&if [null %lut%] &then &do
  &type Usage: LOOKUP <poly_coverage> <lookup_table>
  &return &warning
&end

ae
edit %poly_cov%
ef poly
additem spot 4 12 f 3
sel all
calc spot = range
sel all
LOOKUP range range-code %lut% spot
save
```

```
dropitem %poly_cov%.pat %poly_cov%.pat spot

&return
```

7. polylabel.aml

```
/*****************************************************************
*
/* POLYLABEL - turn contours into range polygons
/* Assumes coverage has been pre-processed to contain a perimeter box
/* and polygon topology.
/* This essentially assigns polygons the median value of their
/* surrounding arcs.
/* A lookup table based on the original contour interval can be used
/* to classify the resulting polygon values.
/**creator of this aml: Unknown
/* Oregon Climate Service
/* Oregon State University
/* Authors: W. Gibson, J. Weisberg
/* (c) March 1998
/*
/* 10/28/98 W. Gibson, T. Parzybok
/* Changed missing from -999 to -9999
/*
/* 1998-2006 W. Gibson
/* Many code changes to fix erroneous labelling errors.  Mostly due to
/* situations in which the raster surface has many small undulations.
/*****************************************************************
*

&args .cp$cover .cp$aat_item .cp$pat_item .cp$pat2_item .cp$pat3_item

/*******************************************************
/* Add a trap for detection of error
/*******************************************************

&severity &error &routine error

ARCPLOT
&if [show program] ne ARCPLOT and [show program] ne GRID &then &do
  &type This program must be run from Arcplot or Grid
  &return &warning
&end

/*****************************************************************
/* check for required arguments and valid 'NET' (line and poly) cover
/*****************************************************************

&if [null %.cp$cover%] &then &do
  &type Usage: CONTOURPOLY <in_cover> {in_contour_item}
{out_range_item}
  &return &warning
&end

&if ^ [exists %.cp$cover% -network] &then &do
```

```
  &type Input cover not found or doesn't contain both line and poly
attributes
  &return &warning
&end

/*********************************
/* check for existance of AAT item
/*********************************

&if [null %.cp$aat_item%] or %.cp$aat_item%_ eq #_ &then
  &set .cp$aat_item := contour

&if ^ [iteminfo %.cp$cover%.aat  -info %.cp$aat_item% -exists] &then
&do
  &type Input contour item [upcase %.cp$aat_item%] not found
  &return &warning
&end

/*************************************
/* check for and add PAT item for range
/*************************************
/*&setv pass = 1

/*&if %pass% = 0 &then  &do /*testing only

&if [null %.cp$pat_item%] or %.cp$pat_item%_ eq #_ &then
  &set .cp$pat_item := range

&if ^ [iteminfo %.cp$cover%.pat -info %.cp$pat_item% -exists] &then
  ARC ADDITEM %.cp$cover%.pat %.cp$cover%.pat %.cp$pat_item% 4 12 f 3

&else &do
  &type Output range item [upcase %.cp$pat_item%] already exists.
  &return &warning
&end

/*****************************************************************
/* check for and add PAT item to flag equal max and min contours
/*****************************************************************

&if [null %.cp$pat2_item%] or %.cp$pat2_item%_ eq #_ &then
  &set .cp$pat2_item := edge

&if [null %.cp$pat3_item%] or %.cp$pat3_item%_ eq #_ &then
  &set .cp$pat3_item := mask

&if ^ [iteminfo %.cp$cover%.pat -info %.cp$pat2_item% -exists] &then
  ARC ADDITEM %.cp$cover%.pat %.cp$cover%.pat %.cp$pat2_item% 4 12 f 3

&else &do
  &type Output range item [upcase %.cp$pat2_item%] already exists.
  &return &warning
&end

/*&end /* temporary check for pass

/* temporary set of variables for pass = 1
```

```
/*&if %pass% = 1 &then  &do /*testing only
/*  &set .cp$aat_item := contour
/*  &set .cp$pat_item := range
/*  &set .cp$pat2_item := edge
/*  &set .cp$pat3_item := mask
/*&end



/*********************************
/* find number of polygons in cover
/*********************************

&setvar missing = -9999
ASELECT %.cp$cover% poly
&set npol := [extract 1 [show select %.cp$cover% poly]]

/*****************************************************************
*
/* get arcs surrounding each polygon
/* except for peaks and pits these will have upper and lower elevations
/* assign mid elevation to polygon
/*****************************************************************
*

/*&if %pass% = 0 &then  &do /*testing only

&do i := 2 &to %npol% /* start from 2 to exclude universe poly

ASELECT %.cp$cover% poly
RESELECT %.cp$cover% poly $recno = %i%

STATISTICS %.cp$cover% poly
   max %.cp$pat3_item%
end

&set maskv := [show statistic 1 1]

 &if %maskv% = 0 &then &do
  CALC %.cp$cover% poly %.cp$pat_item% = %missing%
  CALC %.cp$cover% poly %.cp$pat2_item% = 0
 &end

 &else &do

  ASELECT %.cp$cover% line
  RESELECT %.cp$cover% line lpoly# = %i% or rpoly# = %i%
  RESELECT %.cp$cover% line lpoly# <> 1 and rpoly# <> 1 /* exclude
border segs
  RESELECT %.cp$cover% line contour > %missing%
  &set nrec := [extract 1 [show select %.cp$cover% line]]

  &if %nrec% = 0 &then &do
    &set zmax := %missing%
    &set zmin := %missing%
   &end
```

```
  /*** Peak or pit ***
  &if %nrec% = 1 &then &do

    STATISTICS %.cp$cover% arcs
      max %.cp$aat_item%
    END
    &set val1 := [show statistic 1 1] /* contour value of line
surrounding polygon

    /** Get the adjacent polygon
    STATISTICS %.cp$cover% arcs
      max lpoly#
      max rpoly#
    END
    &set left := [show statistic 1 1]
    &set right := [show statistic 2 1]

    &if %left% = %i% &then
       &set polygon := %right%
    &else
       &set polygon := %left%

    ASELECT %.cp$cover% line
    RESELECT %.cp$cover% line lpoly# = %polygon% or rpoly# = %polygon%
    RESELECT %.cp$cover% line lpoly# <> 1 and rpoly# <> 1 /* exclude
border segs
    RESELECT %.cp$cover% line contour > %missing%

    STATISTICS %.cp$cover% arcs
      max %.cp$aat_item%
      min %.cp$aat_item%
    END
    &set zmx := [show statistic 1 1]
    &set zmn := [show statistic 2 1]

    /** Flat terrain.  will not be able to set the value correctly.
    &if %zmx% = %zmn% &then &do
       &set zmax := %val1%
       &set zmin := %val1%
    &end

    &else &do

       &if %zmx% = %val1% &then &do
         &set zmin := %val1%
         &set zmax := %zmin% + ( %zmx% - %zmn% )
       &end

       &else &do
         &set zmax := %val1%
         &set zmin = %zmax% + ( %zmn% - %zmx% )
       &end

    &end

  &end
```

```
  /*** all the others ***
  &if %nrec% > 1 &then &do
    STATISTICS %.cp$cover% arcs
      max %.cp$aat_item%
      min %.cp$aat_item%
    end
    &set zmin := [show statistic 1 1] /* could catch peaks or pits here
    &set zmax := [show statistic 2 1] /* for special treatment
   &end

  &set val := ( %zmin% + %zmax% ) / 2
  ASELECT %.cp$cover% poly
  RESELECT %.cp$cover% poly $recno = %i%
  CALC %.cp$cover% poly %.cp$pat_item% = %val%
  CALC %.cp$cover% poly %.cp$pat2_item% = 0

 &if %zmax% = %zmin% &then /* flags contours where zmax=zmin
  CALC %.cp$cover% poly %.cp$pat2_item% = 1

/*&if %val% <= 0.0 &then
/*  CALC %.cp$cover% poly %.cp$pat2_item% = 0

/*&if %val% <= 0.0 &then &do
/*&if %val% < 0.0 &then &do
 /*&type How the tarnation did we get here
 /*&type zmin = %zmin%
 /*&type zmax = %zmax%
 /*&type polygon number %i%
 /*&return &error &fatal
 /*&end

 &end

 &end
/*  &return

/*&end

/*********************************************************
/*                           PASS 2                      *
/*********************************************************
/* Reprocess for those polygons that weren't assigned range
/* values correctly in the above code.
/*
/* picks out peaks and adds the difference to their ranges
/*********************************************************

/*&echo &on

&type PASS 2
ASELECT %.cp$cover% poly
RESELECT %.cp$cover% poly edge = 1 and %.cp$pat_item% > %missing%


&sv nrec := [extract 1 [show select %.cp$cover% poly]]
```

```
&type Found %nrec% polygons whose edge was not 0

cursor fixcur declare %.cp$cover% poly rw
cursor fixcur open

&do &while %:fixcur.aml$next%
        &setv i = %:fixcur.$recno%

&type Working on record number %i%

/* ------ Get the current polygon range value

ASELECT %.cp$cover% poly
RESELECT %.cp$cover% poly $recno = %i%

  STATISTICS %.cp$cover% poly
    max range
  end

  &setvar range_cur = [show statistic 1 1]


/* ---- Get lines that are adjacent to current polygon

  ASELECT %.cp$cover% line
  RESELECT %.cp$cover% line lpoly# = %i% or rpoly# = %i%
  RESELECT %.cp$cover% line lpoly# <> 1 and rpoly# <> 1
  RESELECT %.cp$cover% line contour > %missing%

  &set nrec := [extract 1 [show select %.cp$cover% line]]

/* ------- For number of records = 1

  &if %nrec% = 1 &then &do

 STATISTICS %.cp$cover% arcs
   max lpoly#
   max rpoly#
 end

  &setv var_lpoly = [show statistic 1 1]
  &setv var_rpoly = [show statistic 2 1]

/*  &goto jump1

  &end /* %nrec% = 1


/* ------------ For number of records > 2

&if %nrec% > 2 &then &do
    &type Found more than 2 records

/* ----- Get longest line 1st

 STATISTICS %.cp$cover% arcs
    max length
```

```
 end

 sort %.cp$cover% arcs length descending
 &setv l1 = [show select %.cp$cover% line 1 item $recno]
 RESELECT %.cp$cover% line $recno = %l1%

/* &setv l1 = [truncate [show statistic 1 1]]
/* RESELECT %.cp$cover% line length >= %l1%

 STATISTICS %.cp$cover% arcs
   max lpoly#
   max rpoly#
 end
 &setv lp1 = [show statistic 1 1]
 &setv rp1 = [show statistic 2 1]

/* 970331 WPG

  &if %lp1% = %i% &then &do
    &setv var_rpoly = %rp1%
    &setv var_lpoly = %lp1%
  &end

  &else &do
    &setv var_rpoly = %lp1%
    &setv var_lpoly = %rp1%
  &end

/* --- Get smallest inside closed polygon

  ASELECT %.cp$cover% line
  RESELECT %.cp$cover% line lpoly# = %i% or rpoly# = %i%
  RESELECT %.cp$cover% line tnode# = fnode#
  RESELECT %.cp$cover% line lpoly# <> 1 and rpoly# <> 1
  RESELECT %.cp$cover% line contour > %missing%

  &set nsub := [extract 1 [show select %.cp$cover% line]]

 &if %nsub% >= 1 &then &do
  &type tnode is equal to fnode

  STATISTICS %.cp$cover% arcs
     min length
  end

 sort %.cp$cover% arcs length ascending
 &setv l1 = [show select %.cp$cover% line 1 item $recno]
 RESELECT %.cp$cover% line $recno = %l1%


/*  &setv l1 = [truncate [show statistic 1 1]]
/*  RESELECT %.cp$cover% line length <= %l1% + 1

 &end /* nsub >= 1

/* try again for poly#658 fix
```

```
 &if %nsub% = 0 &then &do
  ASELECT %.cp$cover% line
  RESELECT %.cp$cover% line lpoly# = %i% or rpoly# = %i%
  RESELECT %.cp$cover% line lpoly# <> 1 and rpoly# <> 1
  RESELECT %.cp$cover% line contour > %missing%
  RESELECT %.cp$cover% line lpoly# <> %var_rpoly% and rpoly# <>
%var_rpoly%

 &end

  &set nsub := [extract 1 [show select %.cp$cover% line]]

 &if %nsub% <> 0 &then &do

   STATISTICS %.cp$cover% arcs
     max lpoly#
     max rpoly#
   end
   &setv lp1 = [show statistic 1 1]
   &setv rp1 = [show statistic 2 1]

   &if %lp1% = %i% &then
      &setv var_lpoly = %rp1%
   &else
     &setv var_lpoly = %lp1%

  &end /* nsub <> 0

 &end /* %nrec% > 2

/* ------------ For 2 records only

  &if %nrec% = 2 &then &do
    &setv lp1 = [extract 1 [show select %.cp$cover% line 1 item
lpoly#]]
    &setv rp1 = [extract 1 [show select %.cp$cover% line 1 item
rpoly#]]
    &setv lp2 = [extract 1 [show select %.cp$cover% line 2 item
lpoly#]]
    &setv rp2 = [extract 1 [show select %.cp$cover% line 2 item
rpoly#]]

    &if %lp1% = %lp2% and %lp1% = %i% &then &do
        &setv var_lpoly = %rp1% /* lpoly
        &setv var_rpoly = %rp2% /* rpoly
    &end

    &if %rp1% = %rp2% and %rp1% = %i% &then &do
        &setv var_lpoly = %lp1% /* lpoly
        &setv var_rpoly = %lp2% /* rpoly
    &end

    &if %lp1% = %rp2% and %lp1% = %i% &then &do
        &setv var_lpoly = %lp2% /* lpoly
        &setv var_rpoly = %rp1% /* rpoly
    &end
```

```
    &if %lp2% = %rp1% and %rp1% = %i% &then &do
        &setv var_lpoly = %lp1% /* lpoly
        &setv var_rpoly = %rp2% /* rpoly
    &end

  &end /* %nrec% = 2

/* ------ Get the range value for the left polygon

  ASELECT %.cp$cover% poly
  RESELECT %.cp$cover% poly $recno = %var_lpoly%

  STATISTICS %.cp$cover% poly
    max range
  end

  &setv range_lpoly = [show statistic 1 1]

/* ------ Get the range value for the right polygon ---------------

  ASELECT %.cp$cover% poly
  RESELECT %.cp$cover% poly $recno = %var_rpoly%

  STATISTICS %.cp$cover% poly
    max range
  end

  &setv range_rpoly = [show statistic 1 1]

  ASELECT %.cp$cover% poly

/* ---------------------------------------------------------------------
-
/* ----------------- REASSIGN RANGE VALUES IF NECESSARY ---------------
-
/* ---------------------------------------------------------------------
-

/* ----- Current range > range value for left polygon ----------------

  &if %range_cur% > %range_lpoly% &then &do
     &type Current range greater than left
     &sv diff = %range_cur% - %range_lpoly%
/*     CALC %.cp$cover% poly %.cp$pat_item% = %range_cur% + %diff%

     &sv :fixcur.range = %range_cur% + %diff%

  &end

/* ----- Current range > range value for right polygon ----------------
-

  &if %range_cur% > %range_rpoly% &then &do
     &type Current range greater than right
     &sv diff = %range_cur% - %range_rpoly%
/*     CALC %.cp$cover% poly %.cp$pat_item% = %range_cur% + %diff%
```

```
     &sv :fixcur.range = %range_cur% + %diff%

  &end

/* ----- Current range = range for adjacent polygon ------------------
----

  &if %range_cur% = %range_rpoly% and %range_cur% = %range_lpoly% &then
&do

/*     &echo &on
/*     &type Working on record number %i%

/* ---- rpoly is same as current so look left

     &if %i% = %var_rpoly% &then &do
       ASELECT %.cp$cover% line
       RESELECT %.cp$cover% line rpoly# = %var_lpoly% or lpoly# =
%var_lpoly%
       RESELECT %.cp$cover% line lpoly# <> 1 and rpoly# <> 1
       RESELECT %.cp$cover% line contour > %missing%
       RESELECT %.cp$cover% line lpoly# <> %i% and rpoly# <> %i%

       &set nrec := [extract 1 [show select %.cp$cover% line]]

/* ------- Find the line to best use when the above reselect gets more
than 1 line.
/* ------- Assume that the longest line is the best one.
/*--------For nrec = 0 do nothing.  Have to fix by hand (will be caught
by check_poly).

       &if %nrec% > 1 &then &do
        &type More than 1 line so special processing
        STATISTICS %.cp$cover% arcs
         max length
        end

        sort %.cp$cover% arcs length descending
        &setv l1 = [show select %.cp$cover% line 1 item $recno]
        RESELECT %.cp$cover% line $recno = %l1%

       &end

       &if %nrec% >= 1 &then &do

        STATISTICS %.cp$cover% arcs
         max lpoly#
         max rpoly#
        end

        &setv lp1 = [show statistic 1 1]
        &setv rp1 = [show statistic 2 1]

        &if %lp1% = %var_lpoly% &then
         &setv var_cpoly = %rp1%
        &else
         &setv var_cpoly = %lp1%
```

```
        ASELECT %.cp$cover% poly
        RESELECT %.cp$cover% poly $recno = %var_cpoly%

        STATISTICS %.cp$cover% poly
          max range
        end

        &setv new_range = [show statistic 1 1]

        &sv :fixcur.range = %new_range%

        &end /* %nrec% >= 1

        &end

/*  ---- lpoly is same as current so look right

     &if %i% = %var_lpoly% &then &do

        ASELECT %.cp$cover% line
        RESELECT %.cp$cover% line rpoly# = %var_rpoly% or lpoly# =
%var_rpoly%
        RESELECT %.cp$cover% line lpoly# <> 1 and rpoly# <> 1
        RESELECT %.cp$cover% line contour > %missing%
        RESELECT %.cp$cover% line lpoly# <> %i% and rpoly# <> %i%

        &set nrec := [extract 1 [show select %.cp$cover% line]]

/* ------- Find the line to best use when the above reselect gets more
than 1 line.
/* ------- Assume that the longest line is the best one.
/*--------For nrec = 0 do nothing.  Have to fix by hand (will be caught
by check_poly).

        &if %nrec% > 1 &then &do
         &type More than 1 line so special processing
         STATISTICS %.cp$cover% arcs
          max length
         end

         sort %.cp$cover% arcs length descending
         &setv l1 = [show select %.cp$cover% line 1 item $recno]
         RESELECT %.cp$cover% line $recno = %l1%

        &end

        &if %nrec% >= 1 &then &do

         STATISTICS %.cp$cover% arcs
          max lpoly#
          max rpoly#
         end

         &setv lp1 = [show statistic 1 1]
         &setv rp1 = [show statistic 2 1]
```

```
         &if %lp1% = %var_rpoly% &then
          &setv var_cpoly = %rp1%
         &else
          &setv var_cpoly = %lp1%

        ASELECT %.cp$cover% poly
        RESELECT %.cp$cover% poly $recno = %var_cpoly%

        STATISTICS %.cp$cover% poly
          max range
        end

        &setv new_range = [show statistic 1 1]


        &sv :fixcur.range = %new_range%

      &end /* %nrec% >= 1
      &end /* %i% = %var_lpoly%

/* Fix the ones where the current polygon is not the same as either the
right or left

  &if %i% <> %var_rpoly% and %i% <> %var_lpoly% &then &do

  ASELECT %.cp$cover% line
  RESELECT %.cp$cover% line lpoly# = %i% or rpoly# = %i%
  RESELECT %.cp$cover% line lpoly# <> 1 and rpoly# <> 1
  RESELECT %.cp$cover% line contour > %missing%

  STATISTICS %.cp$cover% arcs
    max length
  end

  sort %.cp$cover% arcs length descending
  &setv l1 = [show select %.cp$cover% line 1 item $recno]
  RESELECT %.cp$cover% line $recno = %l1%

  STATISTICS %.cp$cover% arcs
   max lpoly#
   max rpoly#
  end
  &setv fp1 = [show statistic 1 1]
  &setv fp2 = [show statistic 2 1]

  &if %fp1% = %i% &then
    &setv var_cpoly = %fp2%
  &else
    &setv var_cpoly = %fp1%

  ASELECT %.cp$cover% line
  RESELECT %.cp$cover% line rpoly# = %var_cpoly% or lpoly# =
%var_cpoly%
  RESELECT %.cp$cover% line lpoly# <> 1 and rpoly# <> 1
  RESELECT %.cp$cover% line contour > %missing%
  RESELECT %.cp$cover% line lpoly# <> %i% and rpoly# <> %i%
```

```
  &set nrec := [extract 1 [show select %.cp$cover% line]]

  &if %nrec% > 1 &then &do

   STATISTICS %.cp$cover% arcs
     max length
   end

   sort %.cp$cover% arcs length descending

   &setv l1 = [show select %.cp$cover% line 1 item $recno]
   RESELECT %.cp$cover% line $recno = %l1%

   STATISTICS %.cp$cover% arcs
    max lpoly#
    max rpoly#
   end
   &setv fp1 = [show statistic 1 1]
   &setv fp2 = [show statistic 2 1]

   &if %fp1% = %var_cpoly% &then
     &setv var_fpoly = %fp2%
   &else
     &setv var_fpoly = %fp1%

   ASELECT %.cp$cover% poly
   RESELECT %.cp$cover% poly $recno = %var_fpoly%

   STATISTICS %.cp$cover% poly
     max range
   end

   &setv new_range = [show statistic 1 1]

   &sv :fixcur.range = %new_range%

  &end

  &end

/*  &echo &off

  &end


cursor fixcur next

/* end of main loop for PASS 2

&end /*

cursor fixcur close
cursor fixcur remove

/* &echo &off
ASELECT %.cp$cover% poly
```

```
ASELECT %.cp$cover% line

QUIT
&return

/*-----------------------------------------------------
/* Routine ERROR traps for errors and gives the user some
/* idea what polygon is being processed.
/*-----------------------------------------------------

&routine error
&type FATAL ERROR in POLYLABEL.AML
&type Look at file fatal_error for list of current variable definitions
&severity &error &fail
&watch fatal_error
&type An error has occurred in POLYLABEL.AML .....
&type Error occurred on line %aml$errorline%
&type %aml$message%
&type DUMPING variables
&listvar
&watch &off
&return
```

8. put_range.aml

```
/****loop to put latticespot values
/**** of polygon label points to range
/* Oregon Climate Service
/* Oregon State University
/* Authors: W. Gibson, J. Weisberg
/* (c) March 1998
/*
/* 10/28/98 W. Gibson, T. Parzybok
/* Changed missing from -999 to -9999
/*
/* 6/23/05 W. Gibson
/* Changed cell2res := %cellres%  to cell2res := %cellres% /2

&args poly_cov grid_cov cellres

&if [null %poly_cov%] &then &do
  &type Usage: PUT_RANGE <poly_coverage> <grid_coverage> <cellres>
  &return &warning
&end

&if [null %grid_cov%] &then &do
  &type Usage: PUT_RANGE <poly_coverage> <grid_coverage> <cellres>
  &return &warning
&end

&if [null %cellres%] &then &do
  &type Usage: PUT_RANGE <poly_coverage> <grid_coverage> <cellres>
  &return &warning
&end

/*&setv cell2res := %cellres%
&setv cell2res := %cellres% / 2
```

```
&setv missing := -9999

/*************************
/* add these required items
/*************************

&if [index [iteminfo %poly_cov% -polygon range -exists] .FALSE.] gt 0
&then
additem %poly_cov%.pat %poly_cov%.pat range 4 12 f 3
/*additem %poly_cov%.pat %poly_cov%.pat edge 4 12 f 3

/*************************
/*create label point coverage
/*************************
arcedit
edit %poly_cov%
ef label
sel all
put label_cov
quit

build label_cov point

/***find number of polys in cover
ARCPLOT
ASELECT label_cov point
&sv npoly := [extract 1 [show select label_cov point]]

&do i := 1 &to %npoly%  /*point cov. has no univ. poly

&type working on record %i%
  ASELECT label_cov point
  RESELECT label_cov point $recno = %i%

  &setv lonlat = [show select label_cov point 1 XY]
  &setv cell := [show cellvalue %grid_cov% %lonlat%]

  &if [INDEX NODATA %cell%] > 0 &then &do
   &type NODATA found at %lonlat%
   &type Searching for nearest valid cellvalue
   &setv cell = %missing%
   &setv lon = [before %lonlat% ,]
   &setv lat = [after %lonlat% ,]

   &setv lat_shift = %lat% + %cell2res%
   &setv cell1 := [show cellvalue %grid_cov% %lon% %lat_shift%]

   &setv lat_shift = %lat% - %cell2res%
   &setv cell2 := [show cellvalue %grid_cov% %lon% %lat_shift%]

   &setv lon_shift = %lon% + %cell2res%
   &setv cell3 := [show cellvalue %grid_cov% %lon_shift% %lat%]

   &setv lon_shift = %lon% - %cell2res%
   &setv cell4 := [show cellvalue %grid_cov% %lon_shift% %lat%]

   &if [INDEX NODATA %cell1%] = 0 &then
```

```
   &setv cell = %cell1%

   &if [INDEX NODATA %cell2%] = 0 &then
   &setv cell = %cell2%

   &if [INDEX NODATA %cell3%] = 0 &then
   &setv cell = %cell3%

   &if [INDEX NODATA %cell4%] = 0 &then
   &setv cell = %cell4%
  &end
  ASELECT %poly_cov% poly
  RESELECT %poly_cov% poly $recno = %i% + 1
  CALC %poly_cov% poly range = %cell%
&end
QUIT
kill label_cov all
&return
```

9. get_interval.aml

```
/*****************************************************************
*
/* GET_INTERVAL - Creates contour_list for the coverage from the lut
and
/* Calculates the interval to be used in LATTICECONTOUR
/* by getting the third record (the first number after 0
/* will be the smallest interval)
/* "SPOT" in the user-chosen lut (in /array1/lut)
/*
/* Oregon Climate Service
/* Oregon State University
/* Authors: W. Gibson, J. Weisberg
/* (c) March 1998
/*
/* Jul 13, 2005 W. Gibson
/* File name that contains the list of contours is now passed in as an
/* argument.
/*****************************************************************

&args lut work_path confile

&if [null %lut%] &then &do
      &type Missing a lut to process in get_interval.aml
      &type Usage: get_interval <lookup table name> <workspace path>
<output ascii file of contours>
      &type Example: get_interval ak_temp.lut
/home/typhoon/jenny/arc/temp/ak contours_west_07
      &return &error
&end

&if [null %confile%] &then &do
      &type Missing file name that contains the list of contours in
get_interval.aml
      &type Usage: get_interval <lookup table name> <workspace path>
<output ascii file of contours>
```

```
      &type Example: get_interval ak_temp.lut
/home/typhoon/jenny/arc/temp/ak contours_west_07
      &return &error
&end

&if [null %work_path%] &then &do
      &type Missing a workspace path in get_interval.aml
      &type Usage: get_interval <lookup table name> <workspace path>
<output ascii file of contours>
      &type Example: get_interval ak_temp.lut
/home/typhoon/jenny/arc/temp/ak contours_west_07
      &return &error
&end

&setv contour_list = %confile%

&type unloading list of SPOT values

&workspace /array1/lut
  TABLES
  sel %lut%
  unload %work_path%/%contour_list% spot DELIMITED INIT
  quit
&workspace %work_path%

/* Open file
&setv fileunit = [open %contour_list% openstatus -read]
  &if %openstatus% <> 0 &then
  &return &warning Error opening file.

/* Read from file
&type reading lines to get variables
&setv line := [read %fileunit% readstatus]

&if %readstatus% <> 0 &then
      &return &warning Could not read file

&do index = 1 &to 3
  &setv val%index% := %line%
  &type val%index% := [value val%index%]
  &setv line := [read %fileunit% readstatus]
&end

/* Close file
  &if [close %fileunit%] <> 0 &then
  &return &warning Unable to close file

/*Get and type the interval

&setv .contour_interval = %val3% - %val2%
&type The interval is %.contour_interval%

&return
```